www.icgst.com

Text Skew Detection and Correction in Printed Text Images Relying on 2D Haar Wavelets

Tanwir Zaman[1], Vladimir Kulyukin[1], Adele Cutler[2]
Department of Computer Science, Utah State University, Logan, UT, USA[1]
Department of Mathematics and Statistics, Utah State University, Logan, UT, USA[2]
tanwir.zaman@aggiemail.usu.edu, vladimir.kulyukin@usu.edu, adele.cutler@usu.edu

Abstract

A text skew detection algorithm is presented for printed text images. The algorithm applies the 2D Haar Wavelet Transform to an input image to compute the horizontal, vertical, and diagonal change matrices. The matrices are binarized and combined into a single matrix of intensity changes. The convex hull algorithm is applied to find a minimum area rectangle bounding the points in the matrix of intensity changes. The text skew is computed as the rotation angle of the bounding rectangle relative to the absolute north at 90 degrees. No constraints are placed on the magnitude of the text skew. The algorithm's performance is compared with the performance of five text skew detection algorithms on 1001 U.S. nutrition label images and 2200 single- and multi-column document images in multiple languages. The experiments indicate that the proposed algorithm detects text skew angles in real time with an accuracy as high or higher than the accuracy of the other five algorithms. To ensure the reproducibility of the results reported in this article, the JAVA source code of the algorithm is made publicly available.

Keywords: *computer vision, text skew detection, OCR, 2D Haar wavelet transform, mobile nutrition management.*

Nomenclature

NL	Nutrition Label
2D HWT	2-Dimensional Haar Wavelet Transform
TSAW	Text Skew Angle Wavelets
DWT	Discrete Wavelet Transform
AED	Average Error Deviation
CE	Correct Estimations
OCR	Optical Character Recognition

1. Introduction

Adequate comprehension of nutrition labels (NL) is essential for healthy diets in that familiarity with NL terms allows consumers to make better decisions on packaged food products [1]. Computer vision can play a key role in the food selection process by providing consumers with real time text analysis of NLs, which will likely engage consumers in proactive nutrition management [2]. In the United States, the standard nutrition label (NL) of a product lists the percentages of human nutrients supplied in the product that are recommended to be met based on a daily diet of 2000 kilocalories (kcal). This is the most common label format and is required for package sizes with more than 40 square inches. This format is recommended unless there is insufficient continuous vertical space to do so, in which case the side-by-side (i.e., split) format is used and a footnote is placed to the right of the label. The label is enclosed within a text box bound with black colored lines. Each individual ingredient is also separated using horizontal black lines. The NL consists mostly of text and numbers as shown in Figure 1.

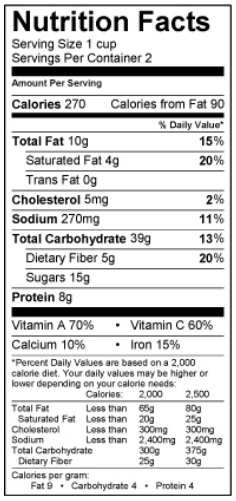

Figure 1. Standard US Nutrition Facts Label

We have developed a vision-based localization algorithm for horizontally or vertically aligned NLs on smartphones [2, 3]. The algorithm was subsequently modified to process not only aligned NLs but also slightly skewed ones. A limitation of the algorithm was its inability to handle arbitrary text skews [4].

In this article, we address this limitation by proposing an algorithm for text skew detection without any constraints on the magnitude of the skew angle. The proposed algorithm works not only on NLs but also on

single- and multi-column printed text images. The algorithm is called TSAW (Text Skew Angle Wavelets) and is implemented in JAVA. To ensure the reproducibility and veracity of the results reported in this article, we have made our source code publicly available [5].

The remainder of our article is organized as follows. In Section 2, related work on text skew detection is reviewed. In Section 3, TSAW is presented. In Section 4, the experiments are described to compare the performance of TSAW with five text skew detection algorithms on 1,001 U.S. NL images and 2,200 document images. The applicability of TSAW in processing real time videos is also evaluated. In Section 5, the results of the experiments are discussed. In Section 6, conclusions are drawn.

2. Related Work

Text skew detection is a well-known problem in computer vision, for which a variety of algorithms have been developed.

Many algorithms calculate text skews with vertical and horizontal projections. A horizontal projection is a 1D array whose size is equal to the number of rows in the image. A vertical projection is a 1D array whose size is equal to the number of columns in the image. The projection profile of an image consists of a 2-tuple of horizontal and vertical projections.

The two projections are computed by rotating an image through a range of angles and calculating black pixels for all rows and columns. At run time, the computed profile of an input image is matched against all precomputed profiles to maximize a specific criterion function. The maximum value of the criterion function determines the text skew.

The concept of projection profiles was pioneered [6] and subsequently patented by Postl [7]. In Postl's algorithm, horizontal projections are computed from 0 to 180 degrees in increments of 5 degrees with the criterion function being the sum of squared differences between adjacent elements of the two projections.

Hull [8] proposes a more efficient text skew detection algorithm similar to Postl's. Unlike Postl's algorithm, Hull's algorithm rotates individual pixels instead of whole images. Specifically, only the coordinates of every black pixel are rotated to reduce space and time to estimate the text skew. Bloomberg et al. [9] improve the efficiency of Postl's and Hull's algorithms by downsampling images prior to computing projection profiles. The researchers propose the variance of black pixel counts in scan lines as the criterion function.

Kanai et al. [10] propose an algorithm that extracts fiducial points in JBIG images by decoding the lowest resolution layer of the JBIG format. The JBIG format includes a progressive encoding method and a lossless compression method for the lowest resolution layer. The fiducial points are projected along parallel lines into accumulator arrays. The text skew is computed as the angle of projection within a search interval that maximizes the alignment of the fiducial points. This algorithm detects skews from ±5 to ±45 degrees.

Li, Shen, and Sun [11] combine projection profiles with wavelet decomposition. Document images are divided into sub-images with the discrete wavelet transform (DWT). The matrix with the absolute values of the horizontal sub-band coefficients is rotated through a range of angles. A step size of 2 degrees is used to compute an initial estimate of the skew angle α. A finer search is then executed from $\alpha - 1$ to $\alpha + 1$ with a step of 0.5 degrees. The algorithm is evaluated on a data set with skews from 0 to ±15 degrees.

Papandreou and Gatos [12] use vertical projections for text skew detection with the criterion function being the sum of squares of the projection elements. This method is claimed to be resistant to noise and image warping and to work best for the languages where most characters include at least one vertical line, which is true for Latin-based languages.

In a more recent publication [13], Papandreou et al. report using minimum bounding box areas of combined horizontal and vertical projection profiles to determine document text skews. They claim that this approach is more resistant to noise and image warp, has no range restrictions on text skews, and is well suited for printed documents.

Shivakumara et al. [14] use linear regression to estimate the skew for each text line segment of a document. Text line boundaries are extracted from projection profiles with static and dynamic thresholds. This algorithm works well for documents with well-separated lines, but loses accuracy for documents with absolute skews greater than 30 degrees.

3. Text Skew Detection

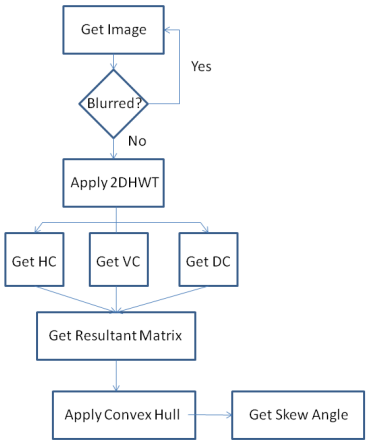

Figure 2. Flowchart depicting the TSAW algorithm

The TSAW algorithm, proposed in this article, uses 2D Haar wavelets to detect text regions in an image and reduce the effective size of the image. Haar Wavelets have been used in image compression techniques. For example, Sudhakar et al. [15] have developed algorithms for image compression based on wavelets.

Al-Abudi et al. [16] present a color image compression scheme based on the Haar Wavelets. Wavelets can also be used for texture classification, as shown by Vijaya Kumar et al. [17].

The TSAW algorithm is shown in Figure 2. TSAW takes printed text images and downsamples them with several iterations of the 2D Haar Wavelet Transform (2D HWT). The HL, LH, and HH matrices are used to identify 2D points with significant horizontal, vertical, and diagonal intensity changes, respectively. These points form a singularity point set in the 2D plane. The convex hull algorithm [18] is applied to this set to enclose it with a minimum area rectangle. The text's skew is the enclosing rectangle's rotation angle relative to the absolute north at 90 degrees.

*A. **Haar Wavelet Transform***

In TSAW, images are downsampled with several iterations of 2D HWT to obtain the HL, LH, and HH matrices. The HWT is a discrete wavelet transform (DWT) applicable to $l^2(Z)$ signals. The recurrences for forward 1D HWT implemented in TSAW are given in (1) and are formally developed in [19].

$$\begin{aligned} d_{j-1}^{(1)}[n] &= s_j[2n+1] - s_j[2n] \\ s_{j-1}^{(1)}[n] &= s_j[2n] + \frac{1}{2} d_{j-1}^{(1)}[n] \\ s_{j-1}[n] &= \sqrt{2}\, s_{j-1}^{(1)}[n] \\ d_{j-1}[n] &= d_{j-1}^{(1)}[n]/\sqrt{2} \end{aligned} \tag{1}$$

The *s* and *d* values are the values of the low and high pass filters, respectively, recursively computed from the previous scale. Unlike more sophisticated DWTs, e.g., Daubechies D4 [20, 21], HWT does not have the boundary problem when the computation of the low and high pass filter values at the current scale requires samples and wavelets outside of the boundaries of $l^2(Z)$ signals.

In TSAW, the generalization of 1D HWT to 2D is based on the tensor products of basic wavelets in the first dimension with basic wavelets in the second dimension, as given in (2). The formal treatment of this generalization is developed in [19].

$$\varphi_{[0,1[}(r) = \begin{cases} 1 \; if \; 0 \le r < 1, \\ 0 \; otherwise. \end{cases}$$

$$\psi_{[0,1[}(r) = \begin{cases} 1 \; if \; 0 \le r < \frac{1}{2}, \\ -1 \; if \; \frac{1}{2} \le r < 1, \\ 0 \quad otherwise. \end{cases} \tag{2}$$

$$\begin{aligned} \Phi_{0,0}^{(0)}(x,y) &= \left(\varphi_{[0,1[} \times \varphi_{[0,1[}\right)(x,y) \\ \Psi_{0,0}^{h,(0)}(x,y) &= \left(\varphi_{[0,1[} \times \psi_{[0,1[}\right)(x,y) \\ \Psi_{0,0}^{v,(0)}(x,y) &= \left(\psi_{[0,1[} \times \varphi_{[0,1[}\right)(x,y) \\ \Psi_{0,0}^{d,(0)}(x,y) &= \left(\psi_{[0,1[} \times \psi_{[0,1[}\right)(x,y) \end{aligned} \tag{3}$$

Given two functions, f_1 and f_2, of one argument, their tensor product is $(f_1 \times f_2)(x,y) = f_1(x) \cdot f_2(x)$. The 2D wavelets for 2D HWT are tensor products of $\varphi_{[0,1[}(r)$ and $\psi_{[0,1[}(r)$ defined in (3), where superscripts *h*, *v*, and *d* denote the horizontal, vertical, and diagonal wavelets, respectively. In 2D $l^2(Z)$ signals, e.g., images, the horizontal wavelets reflect horizontal (left to right) changes, the vertical wavelets reflect vertical (top to bottom) changes, and the diagonal changes reflect the changes between the two main diagonals.

$$\begin{bmatrix} s_{0,0} & s_{0,1} \\ s_{1,0} & s_{1,1} \end{bmatrix} = \begin{bmatrix} 11 & 9 \\ 7 & 5 \end{bmatrix} \tag{4}$$

In practice, 2D HWT is computed by applying 1D HWT to each row and then to each column. As an example, suppose there is a 2×2 pixel image, defined in (4), where $s_{r,c}$ denotes a pixel in row *r* and column *c*. Applying 1D HWT to each row results in the 2×2 matrix in (5). 1D HWT is then applied to each column of the matrix in (5), which results in the matrix in (6) whose coefficients encode the data in the original matrix in (4) in terms of the four tensor wavelets $\Phi_{0,0}^{(0)}(x,y)$, $\Psi_{0,0}^{h,(0)}(x,y)$, $\Psi_{0,0}^{v,(0)}(x,y)$, and $\Psi_{0,0}^{d,(0)}(x,y)$ in (7). This decomposition operation can be represented in terms of matrices in (8).

$$\begin{bmatrix} \frac{11+9}{2} & \frac{11-9}{2} \\ \frac{7+5}{2} & \frac{7-5}{2} \end{bmatrix} = \begin{bmatrix} 10 & 1 \\ 6 & 1 \end{bmatrix} \tag{5}$$

$$\begin{bmatrix} \frac{10+6}{2} & \frac{1+1}{2} \\ \frac{10-6}{2} & \frac{1-1}{2} \end{bmatrix} = \begin{bmatrix} 8 & 1 \\ 2 & 0 \end{bmatrix} \tag{6}$$

$$\begin{bmatrix} 11 & 9 \\ 7 & 5 \end{bmatrix} = 8\Phi_{0,0}^{(0)}(x,y) + 1\Psi_{0,0}^{h,(0)}(x,y) + 2\Psi_{0,0}^{v,(0)}(x,y) + 0\Psi_{0,0}^{d,(0)}(x,y) \tag{7}$$

$$\begin{bmatrix} 11 & 9 \\ 7 & 5 \end{bmatrix} = 8\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + 1\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} + 2\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} + 0\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{8}$$

The value 8 in the upper-left corner of the matrix in (6) is the average value of the original matrix in (4):

(11+9+7+5)/4= 8. The value 1 in the upper right-hand corner of (6) is the horizontal change in the data in (4) from the left average, (11+7)/2=9, to the right average, (9+5)/2=7, which is equal to $1 \cdot \Psi_{0,0}^{h,(0)}(x,y) = 1 \cdot -2$. The value 2 in the bottom-left corner in (6) is the vertical change in the original data in (4) from the upper average, (11+9)/2=10, to the lower average, (7+5)/2=6, which is equal to $2 \cdot \Psi_{0,0}^{v,(0)}(x,y) = 2 \cdot -2 = -4$. The value 0 in the bottom-right corner of (6) is the change in the original data in (4) from the average along the first diagonal (from the top left corner to the bottom right corner), (11+5)/2=8, to the average along the second diagonal (from the top right corner to the bottom left corner), (9+7)/2=8, which is equal to $0 \cdot \Psi_{0,0}^{d,(0)}(x,y)$.

B. Text Skew Detection

TSAW was originally designed to work on NL images taken with smartphone cameras [2], as shown in Figure 3 (left). Interested readers may refer to our previous research [4] on how images with text can be separated from images without text. In the current publicly available implementation [5], the default input image size is $2^{10} \times 2^{10}$.

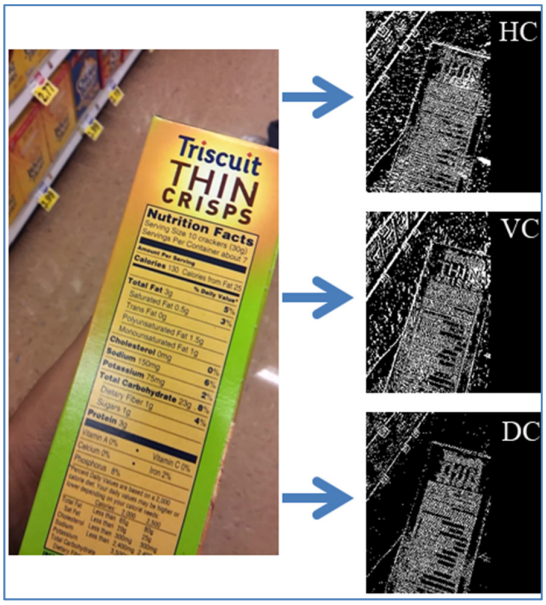

Figure 3. Horizontal, vertical, and diagonal changes

2D HWT is applied to the input image for two iterations to compute the horizontal, vertical, and diagonal changes and store them in the three matrices: HC (horizontal change), VC (vertical change), and DC (diagonal change), as shown in Figure 3 (right). Since 2D HWT is applied twice, the three change matrices are $2^8 \times 2^8$. Thus, the original image is downsampled from 1024×1024 to 256×256. The number of iterations is an input parameter and can be easily adjusted if necessary.

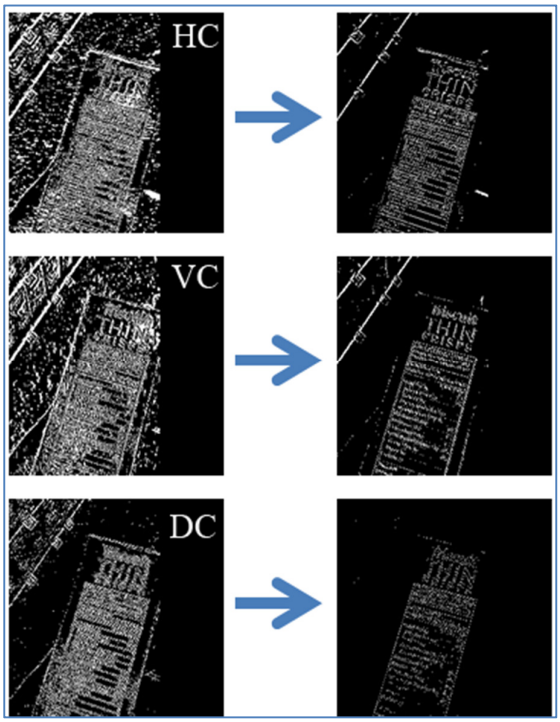

Figure 4. Binarization of HC, VC, and DC matrices

$$S[i,j] = \begin{cases} 255 & \text{if } \alpha HC[i,j] + \beta VC[i,j] + \gamma DC[i,j] \geq \theta \\ 0 & \text{if } \alpha HC[i,j] + \beta VC[i,j] + \gamma DC[i,j] < \theta \end{cases} \quad (9)$$

Each change matrix is binarized to set each pixel to $v_1 = 0$ or $v_2 = 255$, as shown in Figure 4. The binarized matrices are combined into a 256×256 matrix $S[i,j]$ defined in (9), where $\alpha + \beta + \gamma = 1$ and $\theta \in R$ is a threshold. The parameters α, β, and γ control the relative contributions of the horizontal, vertical, and diagonal changes, respectively. A sample $S[i,j]$ matrix is shown in Figure 5 (right) with $\theta = 255$.

The pixels of $S[i,j]$ whose value is 255 indicate 2D points with significant intensity changes. In our previously reported experiments [22], the DC wavelets were observed to detect the presence of text better than the HC or VC wavelets. This may be due to the fact that printed text has more diagonal edges than horizontal or vertical ones as compared to other objects in the image such as lines or graphics. Consequently, in the current implementation of TSAW the following parameter values are used: $\alpha=\beta=0.2$ and $\gamma=0.6$.

The convex hull algorithm [18] is used to find a minimum area rectangle bounding the singularity point set defined by $S[i,j]$, as shown in Figure 6 (right). The text skew is computed as the rotation angle of this rectangle relative to the absolute north at 90 degrees.

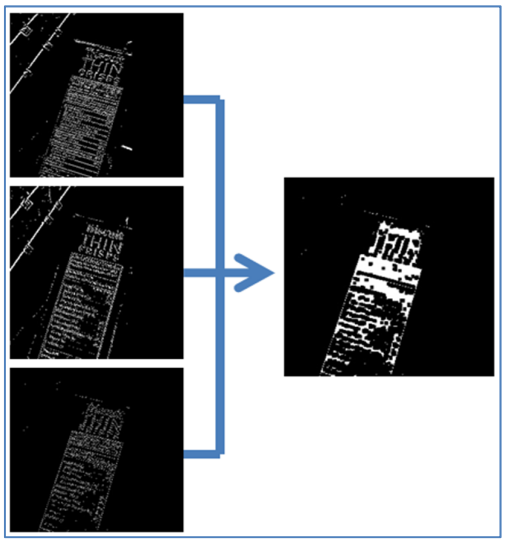

Figure 5. Combining wavelet matrices into result matrix

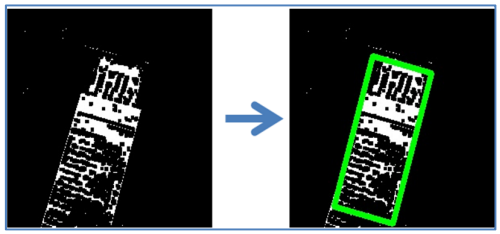

Figure 6. Using the minimum area rectangle for text skew angle computation

Figure 7 gives the TSAW pseudocode. The algorithm takes as input a 2D image *Img* of size $N \times N$, where $N = 2^i, i > 1$. If the size of the image is not equal to an integral power of 2, as required by 2D HWT, the image is padded with 0's. The third argument, *NITER*, specifies the number of iterations for 2D HWT. The algorithm proceeds only if the image is not blurred. This is determined in a pre processing step using our blur detection algorithm [23, 24].

On line 3, 2D HWT is applied to *Img* for *NITER* iterations, which in the current implementation is equal to 2. The procedure 2DHWT returns an array of four n x n matrices *AVRG*, *HC*, *VC*, and *DC*. The first matrix contains the averages while *HC*, *VC*, and *DC* record horizontal, vertical and diagonal wavelet coefficients.

On line 5, the matrices *HC*, *VC*, and *DC* are binarized in place. Lines 7-16 give the pseudocode for the **Binarize** procedure. Figure 4 shows an example of how this procedure works. On line 6, the procedure **FindSkewAngle** is called. The pseudocode for this procedure is shown in lines 17-28. **FindSkewAngle** takes three $n \times n$ matrices *HC*, *VC*, and *DC* and the α, β, γ parameter used in computing $S[i, j]$.

On line 18, the matrix $S[i, j]$ is initialized. On lines 19-27, the $S[r, c]$ values are computed from the $HC[r, c]$, $VC[r, c]$, and $DC[r, c]$ values, as defined in (9). On line 28, the algorithm first calls the procedure **FindMinAreaRectangle** that uses the convex hull algorithm to find a minimal area rectangle around the points with significant intensity changes, i.e., $S[r, c] = 255$. and then calls the procedure **FindRotationAngle** that returns the value of the text skew as the rotation angle of this rectangle relative to the true north of 90 degrees.

```
1.  FUNCTION DetectTextSkewAngle(Img, N, NITER)
2.  If Img is not Blurred Go To step 3 Else return
3.  [AVRG, HC, VC, DC] = 2DHWT(Img, NITER);
4.   n = N/2^NITER;
5.   Binarize(HC, n); Binarize(VC, n);
       Binarize(DC, n);
6.   FindSkewAngle(HC,VC,DC, n);
7.  FUNCTION Binarize(Matrix, n, θ=5, v1=255, v2=0)
8.   For r = 1 to n
9.    For c = 1 to n
10.     If Matrix[r,c] > θ Then
11.       Matrix[r,c] = v1;
12.     Else
13.       Matrix[r,c] = v2;
14.     End If
15.   End For
16.  End For

17.  FUNCTION FindSkewAngle(HC, VC, DC, n, α,β,γ, θ)
18.  Initialize a new n × n S[i,j] matrix with 0's;
19.  For r = 1 to n Do
20.   For c = 1 to n Do
21.     If αHC[r,c] + βVC[r,c] + γDC[r,c] ≥ θ Then
22.       S[r,c] = 255;
23.     Else
24.       S[r,c] = 0;
25.     End If
26.   End For
27.  End For
28.                                   return FindRotationAngle(FindMinAreaRectangle(S));
```

Figure 7. Algorithm's pseudocode

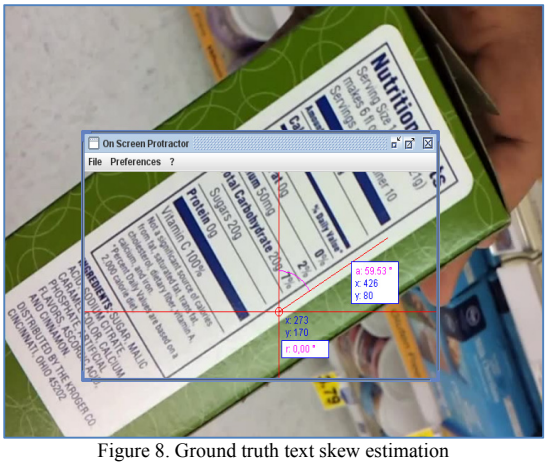

Figure 8. Ground truth text skew estimation

4. Experiments

The performance of TSAW was compared with the algorithms by Postl [6] (Algo1), by Hull [8] (Algo2), by Li, Shen, and Sun [11] (Algo3), and by Papandreou and Gatos [12] (Algo4). Since we were unable to obtain the source code of these algorithms in a performance sensitive imperative programming language (C/C++ or JAVA), we implemented them in JAVA with JDK 1.7, the same JDK version we used to implement TSAW. Our implementations of these algorithms are publicly available [25].

The first experiment was designed to evaluate text skew detection in the context of vision-based nutrition information extraction where the ultimate objective is to extract NL information in real time from NL images taken with smartphones [2, 4]. Toward that end, 1001 NL frames of common grocery products were extracted, at a rate of 1 frame per second, from 1280 x 720 HD videos of common grocery packages with an average duration of 15 seconds. The videos were recorded on an Android 4.3 Galaxy Nexus smartphone in two supermarkets in Logan, UT.

The videos include four different categories of products: bags, boxes, bottles, and cans. Our image blur detection algorithm [23, 24] was used to remove all blurred images from the frames extracted from the videos. Thus, 1001 NL images of common grocery products are the images classified as sharp by our blur detection algorithm. This data set is henceforth referred to as DS1.

The text skew ground truth was obtained from two human volunteers who used an open source protractor [26] to estimate the text skew manually for each image in DS1 as shown in Figure 8. To facilitate the replication of our results, we have made DS1 and the ground truth estimates publicly available [27]. All five algorithms were executed on DS1. The computed text skews were recorded for each image. The text skews were compared with the ground truth via box plots.

In the second experiment, all the algorithms were applied to a public data set (henceforth referred to as DS2) of 2200 scanned document images [28]. DS2 includes figures, tables, diagrams, block diagrams, architectural plans, electrical circuits in multiple languages from newspapers, journals, books, dictionaries, etc. The images are rotated from -5 to +5 degrees with a step of 1. The text skews were logged for each algorithm and image. The text skews were compared with the ground truth via box plots. In box plots, better methods have narrower boxes centered at 0. Wider boxes indicate greater variability between the estimated and actual values. Median lines far away from 0 suggest method biases.

The performance of TSAW was also compared with a more recent algorithm by Papandreou et al. [13]. The researchers evaluated their algorithm on DS2 by computing the average error deviation (AED) and the percentage of correct estimations (CEs) given in (10) and (11), respectively, where N is the number of images in the dataset and $E(j)$ is the error in skew angle estimation for the j-th image. The AED of 0 and percentage CE of 100 are ideal. The AED and CE values were computed for all algorithms on DS1 and DS2.

$$\text{AED} = \frac{\sum_{j=1}^{N} E(j)}{N} \tag{10}$$

$$\text{CE} = \frac{\sum_{j=1}^{N} K(j)}{N} * 100 \text{ , where } K(\text{j}) = \begin{cases} 1 \; if \; E(j) = 0 \\ 0 \; otherwise \end{cases} \tag{11}$$

To test their applicability on real time video processing, we conducted two set of experiments to calculate the average processing times for all the five algorithms. The first set of experiments was carried out on the DS1. All the algorithms were run one by one on the entire dataset of 1001 NL images and the starting and finishing times in milliseconds were recorded. In a second set of experiments the run times of all the algorithms were recorded on DS2 which comprised of 2200 images.

We have also evaluated the performance of TSAW in terms of processing video streams at different frame rates. We took fifty video clips of common grocery products captured with a smartphone in a supermarket. Each video clip has a duration of ten seconds. Then we extracted frames from each clip at 1, 2, and 3 frames per second to obtain three data sets. We ran TSAW on all the extracted frames and recorded the processing times. We also took Algo3, the fastest of the other algorithms investigated in this study and ran it on the same set of video clips after extracting 1,2 and 3 frames per second. We recorded the processing times on each set.

5. Results

In Figure 10, the box plots are given for each algorithm on DS1.The vertical axis denotes the difference between the text skews computed by each algorithm and ground truth. The box plots indicate that TSAW has the narrowest boxes and median errors close to 0 in all image categories, which suggests that this algorithm is less error prone and more consistent than the other four algorithms compared with TSAW on DS1.

Algo3 is a close second with the median errors close to 0. However, the Algo3 boxes are wider than TSAW's. Algo1 has a negative bias for cans, boxes and bottles. Algo2 also has a negative bias for boxes and wider spreads than TSAW in all image categories. While Algo4 has median errors at 0 in all four image categories, it has wider spreads than either Algo3 or TSAW.

The main causes of failure for DS1 were light reflections and irregular product shapes. For example, Figure 9 shows an image of a can on which all algorithms had deviations. The ground truth text skew


on the image in Figure 9 is 66.29 degrees. The skew angles estimated by Algo1, Algo2, Algo3, Algo4 and TSAW on the image in Figure 9 were 60, 0, 90, 120, 77.52, respectively. For TSAW, the light reflections both above and inside the NL caused the point outliers and the subsequent error in the minimum area rectangle identification.

Figure 9. Text skew angle detection error for TSAW

The box plots in Figure 11 indicate that all algorithms performed well on DS2. The horizontal axis denotes the difference between the text skews computed by each algorithm and ground truth. Algo3 has a 0 median error while Algo1, Algo2, and Algo4 have slight negative biases with a median of -1. TSAW has a very narrow box with the median at 0.073 and Q1 and Q3 at 0 and 0.82, respectively. The results of the experiments on DS2 show that although TSAW was originally designed to work in real time on NL images it performs on par with the algorithms designed to detect text skew in standard text documents.

Table 1. AED and CE statistics on DS1

Algorithm	AED	CE
Algo1	67.85	3.50
Algo2	52.96	4.80
Algo3	21.44	9.59
Algo4	45.69	5.89
TSAW	8.60	24.98

Table 2. AED and CE statistics on DS2

Algorithm	AED	CE
Algo1	4.20	55.36
Algo2	6.76	46.09
Algo3	6.33	51.59
Algo4	8.28	43.18
Algo5	0.06	74.50
TSAW	6.11	51.18

Tables 1 and 2 show the AED and CE statistics computed for DS1 and DS2. In Table 2, Algo5 refers to a more recent version of the text skew algorithm by Papandreou et al. [13]. Recall that the ideal values are 0 and 100, respectively.

TSAW has the lowest AED and highest CE on DS1. On DS2 Algo5 has the lowest AED and highest CE values. AED

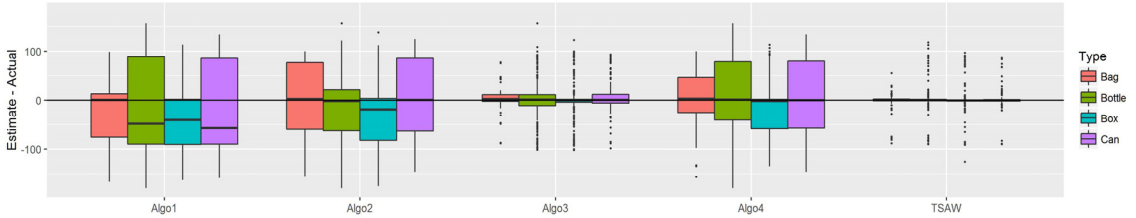

Figure 10. Algorithm box plots on DS1

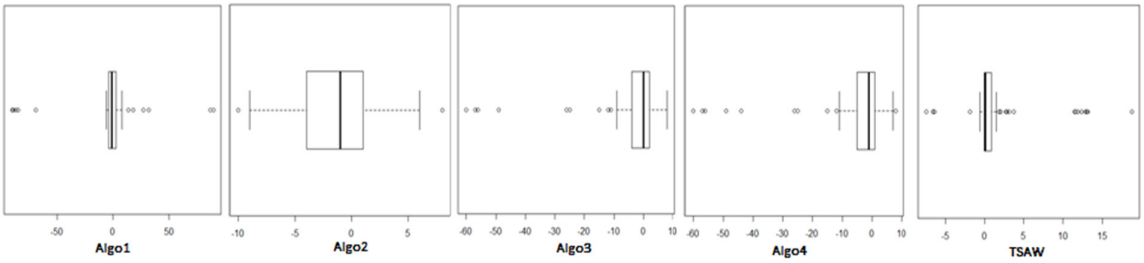

Figure 11. Algorithm box plots on DS2

and CE values on DS2 for TSAW are comparable to the other

algorithms. The results of the experiments on DS2 show that, although TSAW was originally designed to work in real time on NL images, it performs on par with the algorithms designed to detect text skew in printed text documents.

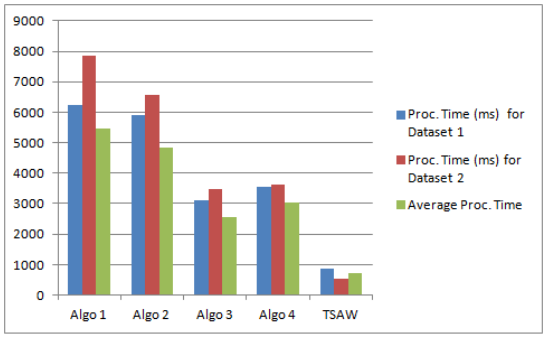

Figure 12. Processing times of the algorithms

Figure 12 shows the processing times for all the algorithms on DS1 and DS2. The vertical axis gives the times in milliseconds. The horizontal axis denotes the different algorithms. It can be seen that TSAW has the least run time of all the algorithms, and is the only one with a run time of less than a second per image. The other algorithms have processing times of more than 3 seconds per frame.

This difference in run time performance is due to the fact that TSAW reduces the image size by a factor of 4 (two 2D HWT iterations) and performs no image rotations to determine the skew. Thus, the overall worst case time complexity of TSAW $O(n^2)$ on n x n input images. The other algorithms have the worst case time complexity of $r \cdot O(n^2)$, where r is the range of angles of rotations, e.g., from -180 to +180 degrees in increments of 5 degrees. In TSAW, on the other hand, there is no overhead associated with repeated image rotation.

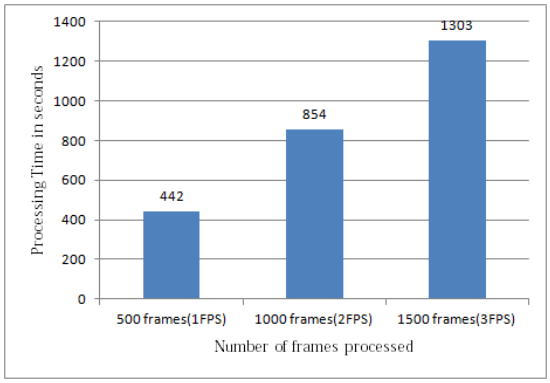

Figure 13. Average processing times (s) for TSAW on videos at different frames rates

Figure 13 shows the processing times for TSAW on video clips with different frame rates. In the first data set obtained after extracting frames at a rate of 1 frame per second, 500 frames were processed by TSAW in 442 seconds. In the second data set, 1000 frames were extracted from the videos at a rate of 2 frames per second and were processed by TSAW in 854 seconds. In the third data set 1500 frames were extracted from the videos at a rate of 3 frames per second, and processed by TSAW within 1303 seconds. In other words, each frame was processed by the algorithm under a second, which makes TSAW usable for real time video processing.

The other algorithms, i.e., Algo1, Algo2, Algo3 and Algo4, are not applicable for real time video processing, because, as can be seen from Figure 12, each of them takes at least three seconds to process one frame. Thus, the performance of the fastest algorithm, i.e., Algo3, is expected to be three times slower than the performance of TSAW.

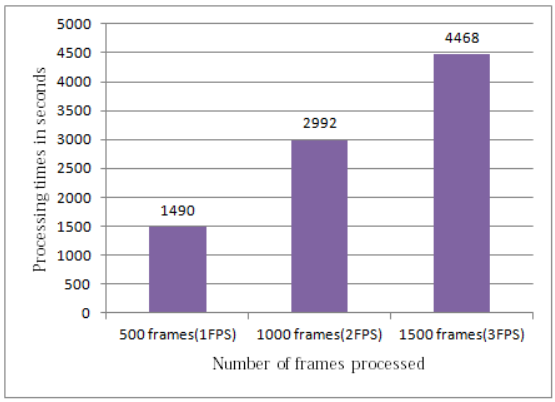

Figure 14. Average processing times (s) for Algo3 on videos at different frames rates

Figure 14 shows the processing times for Algo3 on video clips with different frame rates. We used the same datasets of extracted frames from video clips that were used in the TSAW video experiments in Figure 13. As expected, Algo3 took 1490s, 2992s and 4468s to process the first, second, and third datasets, respectively. In other words, TSAW was three times faster than Algo3 on the video clips.

6. Conclusions

A text skew detection algorithm, called TSAW, is presented that does not place any restrictions on text skew magnitudes. Although TSAW is originally designed to work with nutrition label images captured with handheld mobile phone cameras, it can also detect text skews in printed document images which have significantly better lighting and exposure than NL images. TSAW downsamples text images through several iterations of 2D HWT. The HL, LH, and HH matrices are used to identify 2D points with significant intensity changes. The convex hull algorithm [6] encloses these points it with a minimum area rectangle. The text's skew is the enclosing rectangle's rotation angle relative to the true north.

The performance of TSAW was evaluated on two data sets and compared with the performance of the algorithms by Postl [6, 7] (Algo1), by Hull [8] (Algo2), by Li et al. [11] (Algo3) and by Papandreou et al. [12] (Algo4) and its more recent version [13] (Algo5). The first data set (DS1) consisted of 1001 images of NL extracted from videos captured in grocery stores with

a smartphone camera. The second data set (DS2) consisted of 2200 scanned document images of single- and multi-column documents.

On DS1, TSAW was found to be the most accurate with a median error of 4.62 as compared to 68.85, 20.92, 9.71, and 17.5 for Algo1, Algo2, Algo3, and Algo4, respectively. All the algorithms performed well on images from DS2, which indicates that even though TSAW is originally designed for NL images it performs on par with the algorithms specifically designed to detect text skews in document images.

TSAW was compared with Algo3, the fastest of the other algorithms investigated in the reported study, on video clips with three different frame rates. On all video clips, TSAW was, on average, three times faster than Algo3.

To ensure the reproducibility and veracity of the results reported in this article, we have made publicly available our JAVA source code and a database of NL images annotated with human ground truth estimates [5, 25, 27].

Our future work will be aimed towards coupling the output of TSAW with OCR engines to extract text from localized NLs. In our previous work, a greedy spellchecking algorithm was presented to correct OCR errors in vision-based NL scanning [29]. However, improvements in text skew detection may eliminate the need for spellchecking altogether without lowering the OCR rates.

7. Acknowledgements

We would like to thank Sarbajit Mukherjee for helping us in analyzing the data from the text document experiments.

8. References

[1] Sinclair S, Hammond D, and Goodman S. "Sociodemographic differences in the comprehension of nutritional labels on food products." J Nutr Educ Behav. 45(6), pp. 767–72, 2013.

[2] Kulyukin, V., Kutiyanawala, A., Zaman, T, and Clyde, S. "Vision-based localization & text chunking of nutrition fact tables on android smartphones." In Proceedings of the International Conference on Image Processing, Computer Vision, & Pattern Recognition (IPCV 2013), pp. 314-320, ISBN 1-60132-252-6, CSREA Press, Las Vegas, NV, USA.

[3] Kulyukin, V., Sudini, V. R., Wengreen, H., and Day, J. "A Cloud-based infrastructure for caloric intake estimation from pre-meal videos and post-meal plate waste pictures." In Proceedings of the 19th International Conference on Health Informatics and Medical Systems (HIMS 2015), pp. 161-166, July 27-30, 2015, Las Vegas, NV, USA, CSREA Press, ISBN: 1-60132-416-2.

[4] Kulyukin, V. and Blay, C. "An algoritm for mobile vision-based localization of skewed nutrition labels that maximizes specificity." In Proceedings of the 18th International Conference on Image Processing and Pattern Recognition (IPCV 2014), pp. 3-9, July 21-24, 2014, Las Vegas, NV, USA, CSREA Press, ISBN: 1-60132-280-1.

[5] JAVA source code of the TSAW algorithm: https://github.com/tanwirzaman/HaarTextSkewDetection.

[6] Postl, W. "Detection of linear oblique structures and skew scan in digitized documents." In Proceedings of International Conference on Pattern Recognition, pp. 687-689, 1986.

[7] Postl, W. " Method for automatic correction of character skew in the acquisition of a text original in the form of digital scan results." US Patent 4,723,297,1988. https://www.google.com/patents/US4723297.

[8] Hull, J.J. "Document image skew detection: survey and annotated bibliography.WS" In J.J. Hull, S.L. Taylor (eds.), Document Analysis Systems II, World Scientific Publishing Co., 1997, pp. 40-64.

[9] Bloomberg, D. S., Kopec, G. E., and Dasari, L. "Measuring document image skew and orientation." Document Recognition II (SPIE vol. 2422), San Jose, CA, February 6-7, 1995, pp. 302-316.

[10] Kanai, J. and Bagdanov, A. D. "Projection profile based skew estimation algorithm for JBIG compressed images." International Journal on Document Analysis and Recognition, 1(1), pp.43-51, 1998.

[11] Li, S.T., Shen, Q.H., and Sun, J. "Skew detection using wavelet decomposition and projection profile analysis." Pattern Recognition Letters, 28(5), pp. 555–562, 2007.

[12] Papandreou, A. and Gatos, B. "A novel skew detection technique based on vertical projections." In Proc. of International Conference on Document Analysis and Recognition (ICDAR), pp. 384-388, Sept. 18-21, 2011, Beijing, China.

[13] Papandreou, A., Gatos, B., Perantonis, S. J., and Gerardis, I. "Efficient skew detection of printed document images based on novel combination of enhanced profiles." Int. J. Doc. Anal. Recognit. 17(4), pp. 433-454, 2014.

[14] Shivakumara, P., Hemantha Kumar, G. ., Guru, D. S., and Nagabhushan, P. "Skew estimation of binary document images using static and dynamic thresholds useful for document image mosaicing." In Proc. of National Workshop on IT Services and Applications (WITSA 2003), pp.51-55, Feb 27–28, New Delhi, India, 2003.

[15] Sudhakar, R.,Karthiga, R. and Jayaraman,S. "Image compression using coding of wavelet coefficients – a survey," In Proceedings of Graphics, Vision and Image Processing GVIP, July, 2006,Volume 5, pp. 25-38.


[16] Al-Abudi, B.Q. and George, L.A. "Color Image Compression Using Wavelet Transform." In Proceedings ICGST Conference on Graphics, Vision and Image Processing, GVIP-05,Cairo, Egypt, December, 2005,12,pp. 35-41.
[17] Vijaya Kumar V., Raju U. S. N., Chandra Sekaran,K., Krishna,V. V. "An Innovative Technique of Texture Classification and Comparison Based on Long Linear Patterns Using Wavelets." Graphics, Vision and Image Processing GVIP, October, 2008,Volume 10, pp. 13-21.
[18] Freeman, H. and Shapira, R. "Determining the minimum-area encasing rectangle for an arbitrary closed curve." Comm. ACM, 1975, pp.409-413.
[19] Nievergelt, Y. Wavelets made easy. Birkhauser, Boston, USA, 2001.
[20] Jensen, A. and la Cour-Harbo, A. Ripples in mathematics: the discrete wavelet transform. Springer-Verlag, New York, 2001.
[21] Daubechies, I. and Sweldens, W. "Factoring wavelet transforms into lifting steps." J Fourier Anal. App. 4(3): 245-267, 1998.
[22] Zaman, T., and Kulyukin, V. "Text Skew Angle Detection in Vision-Based Scanning of Nutrition Labels." In Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV 2015), pp. 139-144, July 27-30, 2015, Las Vegas, NV, USA, CSREA Press, ISBN: 1-60132-404-9.
[23] Kulyukin, V. and Andhavarapu, S. "Image blur detection with 2D haar wavelet transform and its effect on skewed barcode scanning." In Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV 2015), pp. 125-131. Las Vegas, NV, USA, CSREA Press. ISBN: 1-60132-404-9.
[24] JAVA source code of image blur detection algorithm described in reference [21]. https://github.com/saratkiran/BlurDetection.
[25] JAVA source code of text skew dection algorithms: https://github.com/tanwirzaman/TextSkewDetectionAlgorithms/.
[26] Open source onscreen protractor program. http://sourceforge.net/projects/osprotractor/.
[27] Dataset 1 (DS1): Online annotated database for NL images. https://usu.box.com/s/9zk660t5h1g0dmw4pjj1x1yp6r7zovp3.
[28] Dataset 2 (DS2): Online database of text documents. https://www.iit.demokritos.gr/~alexpap/dataset_A.rar.
[29] Kulyukin, V., Vanka, A., and Wang, H. "Skip trie matching: a greedy algorithm for real-time OCR error correction on smartphones." International Journal of Digital Information and Wireless Communication (IJDIWC): 3(3), pp. 56-65, 2013. ISSN: 2225-658X.

Biographies

Tanwir Zaman is pursuing his PhD. in Computer Science from Utah State University under the supervision of Vladimir Kulyukin. His research interests are computer vision, mobile and distributed Computing.

Vladimir A. Kulyukin is currently an Associate Professor of Computer Science at Utah State University. Kulyukin's research focuses on wavelet algorithms for video and audio $l^2(Z)$ signals.

Adele Cutler is currently Professor of Statistics at Utah State University. Her main research interests are in statistical learning. She introduced Archetypal Analysis and helped Leo Breiman develop the original implementation of Random Forests.